



Bull, D., & Redmill, DW. (1996). Optimization of image coding algorithms and architectures using genetic algorithms. *IEEE Transactions on Industrial Electronics*, 43(5), 549 - 558.
<https://doi.org/10.1109/41.538612>

Peer reviewed version

Link to published version (if available):
[10.1109/41.538612](https://doi.org/10.1109/41.538612)

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

Optimization of Image Coding Algorithms and Architectures Using Genetic Algorithms

David R. Bull, *Member, IEEE*, and David W. Redmill

Abstract— This paper addresses the application of genetic algorithm (GA)-based optimization techniques to problems in image and video coding, demonstrating the success of GA's when used to solve real design problems with both performance and implementation constraints. Issues considered include problem representation, problem complexity, and fitness evaluation methods. For off-line problems, such as the design of two-dimensional filters and filter banks, GA's are shown to be capable of producing results superior to conventional approaches. In the case of problems with real-time constraints, such as motion estimation, fractal search, and vector quantization codebook design, GA's can provide solutions superior to those reported using conventional techniques with comparable implementation complexity. The use of GA's to jointly optimize algorithm performance in the context of a selected implementation strategy is emphasized throughout and several design examples are included.

Index Terms—Image coding, video coding, genetic algorithms, digital filters, motion estimation, fractal coding, vector quantization.

I. INTRODUCTION

THE requirements for efficient image and video coding techniques increase daily, in line with the growth of the telecommunications and consumer products industries. Applications range from digital broadcast (for both studio and consumer) and cable and satellite delivery systems, through video conferencing and computer-based multimedia to wireless applications. All of these share the requirement for algorithms which faithfully code the original picture data for transmission (or storage) at reduced bit-rates across bandwidth-limited channels.

A block diagram showing the conventional approach to image coding is given in Fig. 1(a). This shows the primary components in the coding process. The image is initially decorrelated (typically using a discrete cosine transform (DCT) operating on subimage regions) in order to reduce the natural interpixel dependence present in most scenes. Alternative techniques include predictive coding, wavelet or subband approaches, fractal coding, and vector quantization. Good overviews of these techniques are given in [1] and [2]. After quantization, the remaining bits are entropy-coded in order to further reduce the transmission bandwidth requirements. The coded information will then be transmitted or stored (experiencing possible corruption by noise and or multipath effects), requiring the addition of channel coding for protection or error-resilient coding methods. The decoder section is

essentially the inverse of the above, ideally producing at its output a perceptually perfect reconstruction of the original input.

In the case of video signals, the codec must deal, not only with a two-dimensional (2-D) spatial domain, but also with a third independent variable: time. Interframe correlation is generally exploited by employing some form of motion-compensated prediction. The transmitted information then simply comprises a set of motion vectors and a coded form of the motion-compensated frame difference signal. This is shown conceptually in Fig. 1(b). In practice, especially at lower bit-rates, the decoded image will inevitably be perceptually degraded. A trade-off thus exists between bit-rate (or compression ratio) and image quality. The aim of any image coder is thus to yield the best quality reconstructed image at the lowest implementation cost, for a given compression ratio.

Genetic algorithms (GA's) have been recognized for some time as a powerful tool for the optimization of difficult problems in engineering design. This paper addresses the application of these techniques to problems in image and video coding and demonstrates their potential when used to solve real design problems with both performance and implementation constraints. GA's embody an adaptive search strategy based on the genetic processes of biological organisms. The foundations of GA's were laid down by Holland [3] and are well described in many texts and tutorial papers (for example, see Beasley, Bull, and Martin [4], [5], Goldberg [6], and other papers in this special section). The power of GA's comes from their robustness and ability to search multimodal, multidimensional search spaces which may be discontinuous or noisy. The success of any GA-based optimization strategy is, however, highly dependent on problem representation, coding, and objective function formulation.

This paper reviews the application and scope of GA techniques to problems in image and video coding. Design problems, ranging from 1-D and 2-D filter and VQ codebook design, to motion compensation, registration, and fractal search are addressed. Section II describes the use of GA's for the design and realization of 1-D and 2-D digital filters. Emphasis is placed on reduced complexity implementations of separable and nonseparable filters for high throughput or low power applications. Extensions of this work to filter-bank implementations for subband and wavelet decompositions are discussed in Section III.

The main reason cited against using GA's for real-time coding applications is their computational complexity and unpredictable convergence characteristics. It is demonstrated in Sections IV and V, however, that for certain real-time image

Manuscript received October 25, 1995; revised November 30, 1995.

The authors are with the Image Communications Group, Center for Communications Research, University of Bristol, Bristol BS8 1TR, U.K.

Publisher Item Identifier S 0278-0046(96)03298-4.

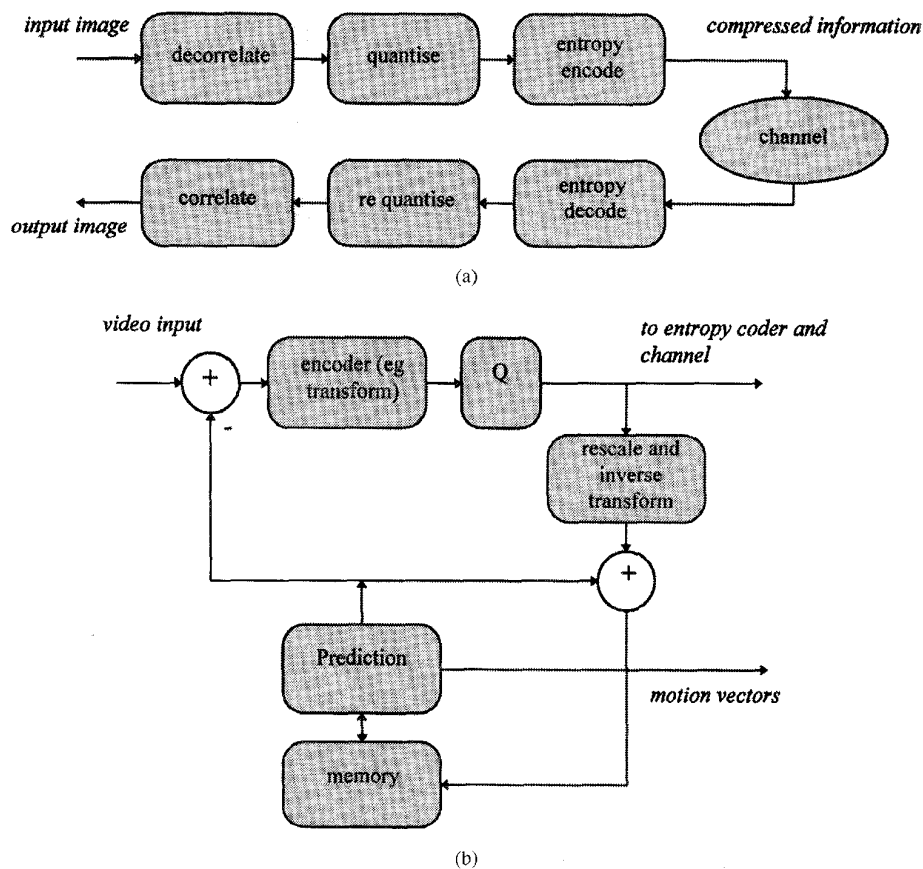


Fig. 1. Basic image and video coding schemes. (a) Image. (b) Video.

coding applications—block matching, motion estimation, and fractal search—algorithms incorporating GA's can compete favorably with the exhaustive or hierarchical search methods conventionally employed. An area closely related to motion estimation is that of image registration. Although not included as a component in existing image coding schemes, this may become important for efficient transmission and storage for multimodal imaging applications such as stereoscopic or healthcare telematic applications. Two further (potentially) real-time applications discussed in Sections VI and VII are object-based coding and the design and searching of codebooks for vector quantization. These are emerging areas with significant potential.

II. DIGITAL FILTER DESIGN FOR IMAGE CODING

A. Introduction

The problem of efficiently realizing high-performance coding algorithms with diverse design constraints is immense. Most design techniques do not cater for broad-based optimization but instead have only a local domain of influence. This is exemplified in filter design, where first a transfer function is developed to satisfy a given frequency response template; then, coefficients are individually (suboptimally) quantized; and finally, decisions are made on architectural aspects (usually with no formal optimization criteria). GA's

have the potential to jointly optimize these diverse and often conflicting constraints.

Applications of 1-D and 2-D digital filters in video and image coding include compression coding, scanning rate conversion, smoothing, and aperture correction [2]. Due to the high computational demands imposed by video digital signal processing (DSP), optimization and complexity reduction methods are in widespread use. In filtering, for example, realization structures generally aim at eliminating the requirement for explicit multipliers and range from the use of very simple structures such as comb filters, through efficient cascades of simple filter sections, to methods which exploit arithmetic redundancy in the quantized coefficient set [7]. Other optimization methods employed to realize efficient fixed-function DSP systems have included linear and dynamic programming [8] and simulated annealing [9].

Many of the above approaches constrain the individual filter coefficients to be simple combinations of signed power-of-two (SPT) terms, thereby enabling more complex multipliers to be replaced by shift and addition/subtraction operations. However, such restrictions imply a discrete and nonuniformly populated solution space which is incompatible with many optimum filter design methods. In the past, this has been overcome either by rounding coefficients generated by optimal continuous designs, or by searching the solution space using simulated annealing [9] or linear programming methods [10]. The rounding approach is known to yield suboptimal filters,

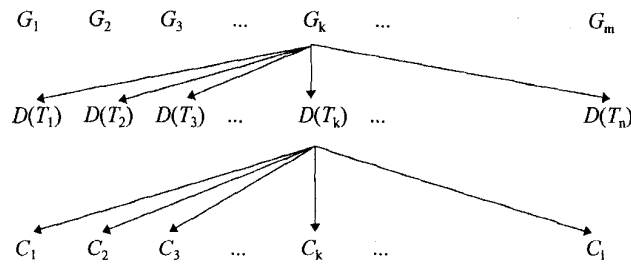


Fig. 2. Structured GA chromosome organization.

while the search-based methods require a large computational effort to effectively search the solution space while not guaranteeing an optimum solution. GA's have previously been identified as a useful tool for efficiently optimizing large discrete multimodal search spaces. This section reviews their use in this context.

B. 1-D FIR Filter Optimization

Due to the normal requirement for phase linearity in image or video coding, one-dimensional (1-D) finite-duration impulse-response (FIR) filters find widespread application. The difference equation for a 1-D FIR filter is given in (1). Here the output $y[n]$ is produced through the convolution of an input sequence $x[n]$ (typically the rows or columns of a 2-D image) with a filter impulse response $h[n]$.

$$y[n] = \sum_{i=0}^N h[i]x[n-i]. \quad (1)$$

GA's have been applied with some success to the design of 1-D digital filters with discrete coefficients. Schaffer and Eshelman have reported good results for multiplier-free filters where coefficients are restricted to a small number of signed power of two terms [11]. Ifeachor and Harris [12] introduce a flexible approach to the design of frequency sampling FIR filters claiming filter performances equal or better to those produced using linear programming with significantly reduced design time.

Recent work by Wade *et al.* [13] describes a method for designing multiplier-free FIR filters comprising an ordered cascade of parameterized linear phase primitive sections. They employ a weighted fitness function which addresses complexity as well as filter performance in the following manner:

$$f = \frac{1}{\sum_j \varepsilon + w_1 \sum_i a_i + w_2 \sum_i d_i} \quad (2)$$

where the first term represents the sum of extremal errors of a given design in the frequency domain and the second two terms, weighted adder and delay counts. A structured GA is adopted which embodies a multilevel chromosome structure (Fig. 2) where higher level genes define the primitive type (T_k) and activate lower level genes containing information such as delay values $D(T_k)$ and power of two coefficient values (C_k). This approach yields efficient albeit high order filters in a low design time compared with linear programming approaches. Typical savings quoted for a variety of filter specifications range up to 50% in chip area.

C. Graph-Based Filter Optimization

The directed graph is well established as a vehicle for algorithmic engineering in all aspects of signal processing [7], [14], [15], including the synthesis of transforms, 1-D and 2-D filters, subband coders, and more general matrix-vector operations. It offers a simple yet flexible means of manipulating DSP algorithms and provides a vehicle for both design and architectural mapping while enabling easy assessment layout regularity, interconnectivity, timing, latency, and arithmetic complexity. In the work of Bull and Horrocks [7], directed graph techniques are employed to optimize the inner product formation process for FIR digital filters.

A directed graph can then be represented in matrix form (3), where $v[n]$ contains the values at each vertex, F_c contains the gains for each internal edge and B contains gains for each edge from a source vertex to each internal vertex. Note that F_c^T must be lower triangular with zero diagonal to ensure a computable graph. An example graph is given in Fig. 3, for coefficient values 17 and 59. The matrix form for this graph is given in (4)

$$v[n] = F_c^T v[n] + Bx[n] \quad (3)$$

$$\begin{bmatrix} v_1[n] \\ v_2[n] \\ v_3[n] \\ v_4[n] \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2^4 + 1 & 0 & 0 & 0 \\ 2^2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 \end{bmatrix} \begin{bmatrix} v_1[n] \\ v_2[n] \\ v_3[n] \\ v_4[n] \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} x[n]. \quad (4)$$

A simple goal of the optimization process might be to obtain the graph using the smallest number of adders and subtractors, with a suitably scaled fitness function, based on some distance measure, used to support this criterion. A GA approach to this problem has been presented by Bull *et al.* in [16]. A two-string chromosome was employed to encode the solutions: the first was used to represent the source vertices for the input edges to each vertex and the second to code the edge gains. For example, the chromosome representing the graph of Fig. 3 would be

$$\begin{aligned} \text{string1} &= (1, 1, 1, 2, 2, 3) \\ \text{string2} &= (00100, 00000, 00010, \\ &\quad 00000, 00000, 00001) \\ &= (2^4, 1, 2^2, 1, 1, 2). \end{aligned}$$

Graph vertices in this case were restricted to comprise two input adders, and graph edges were constrained to powers-of-two values. The standard genetic operators were applied to both strings except that, in the case of the first string, mutation was applied in such a way as to ensure the preservation of a lower triangular matrix and hence a computable graph. In addition to crossover and mutation, a repair operator was also used to form a valid solution where possible from a nonvalid one. The fitness of a candidate solution was defined in order to address both the validity and complexity of solutions. Using a crossover probability of 0.6 and a mutation probability of 0.02 for both strings, with a range of multiplicative coefficients, for different matrix and population sizes, an optimum solution was found in 85% of runs. These figures were compared with a random search (over 100 000 individuals), which yielded a similar solution in only 0.08% of trials.

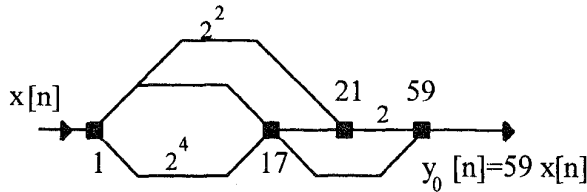


Fig. 3. Directed graph for generating products $17 \times [n]$ and $59 \times [n]$.

An iterative extension of the above algorithm was applied in [16] to the problem of forming a multiple-output directed graph (as would be required by a 1-D FIR filter) using the minimum number of vertices. This was applied to a range of filters including an order 31 linear-phase high-pass FIR filter with the following 8 bit coefficients $\{1, 2, 3, 5, 6, 9, 10, 14, 30, 49, 70, 90, 107, 120, 127\}$. Results, using a support matrix of 35 vertices over 100 trials showed that an optimum (10 adder) solution was found in 44% of the trials.

III. TWO-DIMENSIONAL DIGITAL FILTER DESIGN

A. Filter Design Formulation

A 2-D digital filter with impulse response $h(\mathbf{n})$, $\mathbf{n}^T = (n_1, n_2)$ has a transfer function (5)

$$H(\mathbf{f}) = \sum_{\mathbf{n}} h(\mathbf{n}) e^{-j2\pi \mathbf{f}^T \mathbf{n}}$$

with

$$\mathbf{f}^T = (f_1, f_2) \in \mathbb{R}. \quad (5)$$

To ensure a low-pass linear-phase response the coefficients must obey even symmetry in both the horizontal and vertical directions and for a filter with an odd number of taps in both directions this can be expressed as $h(n_1, n_2) = h(n_1, -n_2) = h(-n_1, n_2) = h(-n_1, -n_2)$ with $n_1 = -N_1, \dots, 0, \dots, N_1$, and $n_2 = -N_2, \dots, 0, \dots, N_2$. If the filters are further constrained to be horizontally and vertically equivalent, then symmetry must also exist about the diagonals; thus, $h(n_1, n_2) = h(n_2, n_1)$ and $N_1 = N_2 = N$. Coefficient values therefore need only be specified for the range $0 \leq n_1 \leq N$ and $0 \leq n_2 \leq n_1$. Coefficients $h(\mathbf{n})$ are assumed to be sums or differences of two power-of-two terms, i.e.,

$$h(\mathbf{n}) \in D$$

$$D = \left\{ \alpha : \alpha = \sum_{k=1}^2 c_k \cdot 2^{-g_k} \right\}$$

$$c_k \in \{-1, 0, 1\},$$

$$g_k \in \{1, 2, \dots, B\} \quad (6)$$

where B denotes the maximum shift value used. Since the filter coefficient space is discrete and sparsely populated, it is useful to incorporate a gain term G in the transfer function. For the octal symmetry conditions the transfer function then becomes as in (7) at the bottom of this page. If $T(\mathbf{f})$ is the desired frequency response of the filter and $W(\mathbf{f})$ is a weighting function, the cost function of the minimax design process, for a 2-D multiplierless FIR filter specified by $G, \{h(\mathbf{n})\}$ can be formulated as in (8), where \mathbf{f}_k represents a predefined sampling grid for evaluating (7)

$$\delta(G, \{h(\mathbf{n})\}) = \max_{\mathbf{f}_k} W(\mathbf{f}) |H(\mathbf{f}) - T(\mathbf{f})|. \quad (8)$$

B. Circularly Symmetric Low-Pass Filters

In [17] and [18], Sriranganathan *et al.* encode each coefficient in the range $0 \leq n_1 \leq N$ and $0 \leq n_2 \leq n_1$ as a gene within the GA, while remaining coefficients are inferred by octal symmetry. Each coefficient consists of 2 terms, $c_k \cdot 2^{-g_k}$. These are specified using 1 bit for the sign of c^k and 3 or 4 bits for the shift value, g^k (depending on whether $B > 8$). The value of G is inferred from the normalized ripple characteristics of the final filter.

The use of restricted signed power of two coefficients has been extended by the authors to allow an unrestricted allocation of SPT terms to individual coefficients while still maintaining a constraint on the total number of terms available in the design process. This has yielded greater improvements in filter performance, due to the ability of the GA to allocate terms according to individual coefficient sensitivities. Table I shows the results of this work compared with those given in [9], [10] using SA and mixed integer linear programming (LP) with identical constraints on the filter coefficients. It should be noted that in the case of LP, filters B, C, and H represent designs to B bit finite precision whereas filters A, D, and G are the result of 2-SPT optimization. Also included are the full precision minimax optimization results (opt) and designs obtained by applying the McClellan transform (MCT) to an optimum 1-D solution. For the GA cases, GA1 represents the constrained case allowing, at most, 2 SPT terms per coefficient whereas GA2 represents the unconstrained case with an average of 2 terms per coefficient. An example frequency response plot for the 7×7 filter (C) designed using the GA approach is shown in Fig. 4. As can be observed, GA's are capable of producing results superior to those obtained using competing methods. Even with restricted dynamic range and only two nonzero terms per coefficient, in many cases their performance approaches that

$$H(f_1, f_2) = \frac{1}{G} \left\{ h(0, 0) + \sum_{n_1=1}^N 2h(n_1, 0) [\cos(2\pi n_1 f_1) + \cos(2\pi n_1 f_2)] + \sum_{n_1=1}^N 4h(n_1, n_1) \cos(2\pi n_1 f_1) \cos(2\pi n_1 f_2) \right. \\ \left. + \sum_{n_2=1}^{N-1} \sum_{n_1=n_2+1}^N 4h(n_1, n_2) [\cos(2\pi n_1 f_2) \cos(2\pi n_2 f_1) + \cos(2\pi n_2 f_2) \cos(2\pi n_1 f_1)] \right\}. \quad (7)$$

TABLE I
PERFORMANCE COMPARISONS BETWEEN 2-D MULTIPLIERLESS FIR FILTER DESIGNS USING FULL PRECISION MINIMAX OPTIMIZATION (OPT) [19],
MCLELLAN TRANSFORM DESIGN, LINEAR PROGRAMMING (LP) [10], SIMULATED ANNEALING (SA) [9],
AND GENETIC ALGORITHMS (GA1 AND GA2)

Design	A	B	C	D	E	F	G	H
Size	5x5	5x5	7x7	7x7	7x7	7x7	9x9	9x9
Passband radius, fp	0.2	0.15	0.15	0.15	0.175	0.225	0.2	0.15
Stopband radius, fs	0.3	0.35	0.35	0.35	0.325	0.275	0.3	0.35
B	11	7	7	9	11	11	5	7
δ (opt)	0.26705 -11.5 dB	0.13188 -17.6 dB	0.03284 -29.7 dB	0.03284 -29.7 dB	0.06557 -23.7 dB	0.24733 -12.1 dB	0.11417 -18.8 dB	0.02609 -31.7 dB
δ (MCT)	0.31931 -9.9 dB	0.19512 -14.2 dB	0.08210 -21.7 dB	0.08210 -21.7 dB	0.12402 -18.1 dB	0.32910 -9.7 dB	0.19481 -14.2 dB	0.08206 -21.7 dB
δ (LP)	0.26766 -11.5 dB	0.14264 -16.9 dB	0.06250 -24.1 dB	0.03831 -28.3 dB			0.24466 -12.2 dB	0.05444 -25.3 dB
δ (SA)	0.26816 -11.4 dB	0.13929 -17.1 dB	0.06024 -24.4 dB	0.03760 -28.5 dB	0.07969 -22.0 dB	0.27514 -11.2 dB	0.22310 -13.0 dB	0.05106 -25.8 dB
δ (GA1)	0.26737 -11.5 dB	0.13585 -17.3 dB	0.04045 -27.9 dB	0.03754 -28.5 dB	0.07256 -22.8 dB	0.26364 -11.6 dB	0.14044 -17.1 dB	0.04051 -27.8 dB
δ (GA2)	0.26736 -11.5 dB	0.13330 -17.5 dB	0.03678 -28.7 dB	0.03650 -28.8 dB	0.07110 -23.0 dB	0.26050 -11.7 dB	0.12493 -18.1 dB	0.03505 -29.1 dB

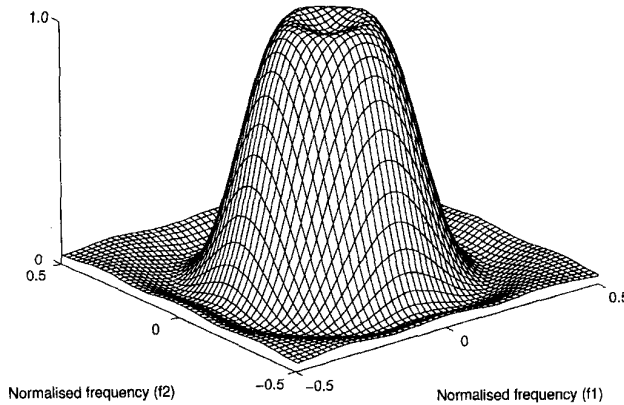


Fig. 4. Frequency response plot for 7x7 filter (C).

TABLE II
FILTER COEFFICIENTS FOR DESIGN C (SCALE FACTOR $G = 4.15148$)

Coefficients	Values	Coefficients	Values
$h(0,0)$	$2^{-1} + 2^{-2} + 2^{-3} + 2^{-7}$	$h(2,2)$	-2^{-4}
$h(1,0)$	$2^{-1} + 2^{-4}$	$h(3,0)$	-2^{-4}
$h(1,1)$	$2^{-2} + 2^{-3}$	$h(3,1)$	-2^{-4}
$h(2,0)$	2^{-3}	$h(3,2)$	-2^{-5}
$h(2,1)$	2^{-5}	$h(3,3)$	-2^{-7}

of the optimum (infinite precision) minimax solution, albeit with greatly reduced implementation complexity. An example set of coefficient values (for filter C in Table I) is presented in Table II.

The size of the search space for this problem, measured in terms of the representation, used ranges from 2^{54} for a 5x5 filter with 7-bit dynamic range to 2^{165} for a 9x9 filter with 11-bit dynamic range. The robustness of the GA-based design method is demonstrated in [18]. Using 20 different random number seeds for filter C the GA yielded an average ripple value of 0.06161 with a best result of 0.0437 and a

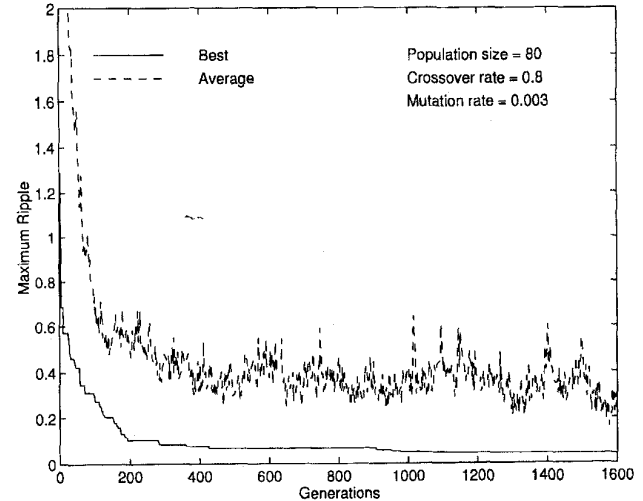


Fig. 5. GA convergence characteristics.

variance of 0.0133. An example of convergence is presented in Fig. 5 which shows both average and best performance of the population in each generation for filter C.

C. Diamond-Shaped Filters for Quincunx Subsampling

A second type of filter considered in [18] is useful when subsampling on a diagonal or quincunx sampling lattice. In order to avoid aliasing, low-pass filtering is necessary prior to subsampling and it can be shown that the optimum half-band low-pass filter has a frequency response of

$$H(f_1, f_2) = 1, \quad f_1 + f_2 < 0.5 \\ 0, \quad f_1 + f_2 > 0.5. \quad (9)$$

This corresponds to a diamond-shaped pass-band in the 2-D frequency domain. For down-sampling it is also useful for the filter to have skew symmetry along the frequency diagonals

$$H(0.5 - f_1, 0.5 - f_2) = 1 - H(f_1, f_2). \quad (10)$$

GA's were used to design this type of filter using an approach similar to that employed for the circularly symmetric case. An example result is shown in Fig. 6. This filter is specified with band edges at normalized frequencies of 0.3 and 0.7. The stopband attenuation achieved for this design, again with the constraint of 2 nonzero digits (over an 11-bit dynamic range) was -32 dB ($\delta = 0.0252$). This compares with a value of -49.59 dB for the same design with 11-bit full precision integer coefficients obtained using a linear programming approach [10] and -27.3 dB for the case of rounding this to 2-SPT.

IV. DIGITAL FILTER BANKS FOR IMAGE CODING

A. Subband Coding

Subband decompositions [1], [20] encompass a large variety of transforms and filter types including: fast Fourier transform (FFT), discrete cosine transform (DCT), overlapped transforms, quadrature mirror (QMF) filters, and discrete wavelet transforms. The latter in particular are finding increased application in coding systems, mainly due to the perceptually preferable nature of the artifacts introduced during compression. In subband image coding, an image is split into a number of different frequency bands obtained by filtering and subsampling the original image. Good compression can be achieved if most of the original information is contained within relatively few subbands.

One important class of subband decomposition, referred to as a wavelet transform, is achieved by a cascade of 2-band systems [Fig. 7(a)] recursively applied to the low band [Fig. 7(b)].

Consider the basic two-band building block shown in Fig. 7(a). This can be characterized by

$$\hat{X}(z) = X(z)\{H_0(z)G_0(z) + H_1(z)G_1(z)\} + X(-z)\{H_0(-z)G_0(z) + H_1(-z)G_1(z)\}. \quad (11)$$

The term $X(-z)$ represents an aliasing component which can be removed when

$$H_1(z) = z^k G_0(-z)$$

and

$$G_1(z) = -(-z^{-k})H_0(-z). \quad (12)$$

An important class of alias cancellation filters are QMF filters which are defined with $H_1(z) = H_0(-z^{-1})$. These define the high-pass filter to be the mirror of the low-pass filter and hence only require the design of one filter. Perfect reconstruction can be achieved by constraining the transfer function to be a pure delay term.

Typical design objectives for this type of system generally represent a trade-off between complexity, frequency response, and reconstruction error. It may at first appear that GA's could be used directly to jointly optimize $H_0(z)$ and $H_1(z)$ to achieve both close to perfect reconstruction and efficient filters. However, for image coding applications a significant problem arises: optimality is generally subjective (and traditional objective measures such as PSNR do not always correlate well with

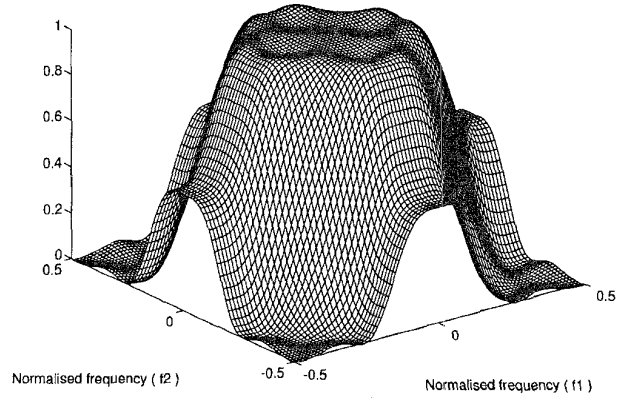


Fig. 6. Frequency response for a 13×13 diamond-shaped filter with $B = 11$.

a subjective assessment of image quality). Clearly, in order to use genetic algorithms, it is necessary to define in some analytic way, what the optimality criteria for a subband filter bank are. Although complexity can be relatively easily defined, the issue of compression quality is clearly more difficult. Useful objective functions might include some combination of frequency response and impulse response characteristics from the individual filters. Other objective quality measures may include the closeness to perfect reconstruction, coding gain or entropy.

It is clear that before GA's can be usefully employed in this area, there are two issues which need to be addressed: the representation to be used and the design objectives. For the problem of subband filter design, neither is well defined. This is an area of ongoing research.

B. Graph-Based Methods for Filter Bank Realization

The above discussion has demonstrated that the design of subband filter coefficients is currently difficult within a GA framework. However, once quantized coefficient values exist, there is generally a requirement to find the most efficient implementation. Bull *et al.* [15] apply a graph based approach, similar to that described in Section II, to the problem of realizing video subband filter banks. This, in combination with a data multiplexing regime which implements a 2-channel QMF using a single FIR structure, facilitated the fabrication of a 64-channel subband coder and decoder on a single gate array. Different filters were specified for horizontal and vertical, analysis and synthesis, and high- and low-pass processing. This example was later redesigned using GA methods [16] similar to those presented in Section II. A support matrix with 50 vertices was employed, and the solution found was superior (in terms of the number of adders-subtractors required for the inner product processes) to that previously published in reference [15]. Although much work still remains to be done in this area, these preliminary results demonstrate the potential of the GA approach.

V. GA'S FOR MOTION ESTIMATION AND IMAGE REGISTRATION

A. Image Registration

In image coding, there is an increasing demand for real-time solutions to complicated and computationally intensive

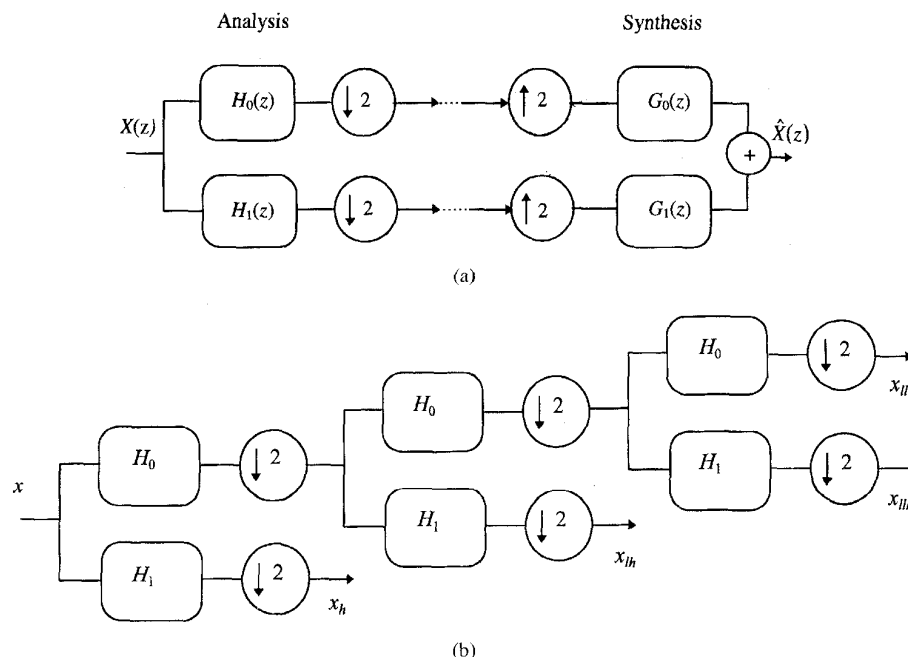


Fig. 7. Subband decomposition. (a) Two-band case. (b) Hierarchical case.

optimization problems, such as image registration and motion estimation. The task of registration is to find the parameters of a transformation that produce the best match between two different images, possibly taken at different times, or using different imaging modalities or from different camera positions. These parameters often represent an affine transformation (spatial shift, rotation, and scaling). Areas of application include medical imaging and remote sensing and compression coding of image sequences, involving camera pan, zoom, or other complex movements. Because the complexity of the search for the transform parameters increases rapidly with transform complexity and image size, GA's represent an attractive potential solution to this problem.

Fitzpatrick *et al.* [21], [22] proposed the use of GA's for medical image registration. They used a stochastic or noisy fitness function [23] approximated by considering only a small, randomly selected set of points. This reduces computational effort at the expense of introducing additional approximation error, which can be viewed as a noise component in the fitness landscape. GA's have been shown to be robust to such noise and, as such, offer efficient solutions to the registration problem. Jacq and Roux [24] used a similar technique for the registration of 3-D-3-D and 2-D-3-D medical images. Tsang *et al.* [25] demonstrate the use of GA's for matching single objects in an image which have been subjected to an affine transformation. His approach, based on matching object contours represented using a polygon approximation, is claimed to be less prone to trapping in local minima and also faster than simulated annealing. For machine vision applications with single objects, convergence was obtained in under 20 generations with a population size of 30.

B. Motion Estimation

A similar problem to image registration is motion estimation [1], [26]. To achieve high compression ratios for video

sequences, it is often useful to track the motion of image regions or objects between successive frames. In this way, the interframe prediction error can be significantly reduced at the cost of generating and transmitting a small number of motion parameters. A commonly employed technique is block matching. Here, each frame is split into square blocks, typically of 16×16 or 8×8 pixels and, for each block, the motion estimator searches for an optimal displacement vector that matches the block in the current frame to a block from the previous frame. Although exhaustive search is feasible, it is computationally expensive. Hence algorithms based on hierarchical approaches, that give a good match with significantly reduced effort are commonly used.

In order to demonstrate the potential of GA's for the block matching application, the authors have simulated a GA based approach and compared the results with those from an exhaustive search and various multi-step algorithms (see Fig. 8). The GA operates on each 16×16 block in the current frame, with chromosomes representing the displacement vectors (quantized to the nearest half pixel). A population size of 25 members was used and the population was seeded with vectors from a 5×5 grid with a spacing of 2 pixels, centered around the origin. With the coding scheme used, the crossover operation was found to be too disruptive and experiments showed good results using mutation only. The mutation operator was implemented as the addition of random Gaussian distributed discrete vectors. Experiments were performed by averaging the mean absolute difference (MAD) of the prediction error for each block in the first 10 frames of the "Claire" sequence (a total of 2560 blocks). Exhaustive search was performed with maximum displacements of 2, 4, 8, and 16 pixels and multistage algorithms were tried with various grid-sizes and spacings. Fig. 9 shows a comparison of the results achieved for exhaustive search, multistep search, and the GA-based search. Although for a very low numbers of

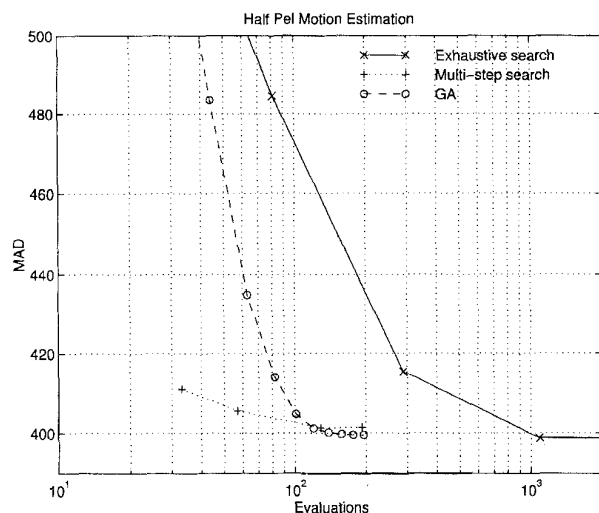


Fig. 8. Performance of GA for block-matching motion estimation.

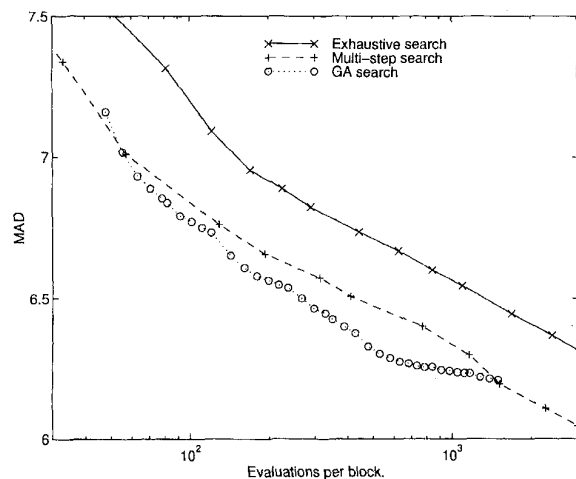


Fig. 9. Performance of GA for fractal search.

evaluations, the multigrid algorithm appears superior, the GA provides better and virtually optimal results for intermediate numbers of evaluations. It is also anticipated that the GA's will offer a feasible solution to more difficult motion estimation problems, using more complicated transformations including rotations and scaling parameters (see Section IV-A). However consideration must be given to the cost (in terms of bit-rate) of transmitting these additional parameters. This is a fruitful area for further study.

VI. FRACTAL IMAGE CODING

Recently, methods of image coding have been proposed based on the theory of iterated function systems (IFS). This approach is commonly referred to as fractal image coding [27] due to the exploitation of self-similarity within the image. The image to be coded is represented as an approximate transformation T of itself. Under certain conditions, it can be shown that the iterated equation $\hat{I}_n = T(\hat{I}_{n-1})$ with

arbitrary \hat{I}_0 will converge to some image $\hat{I} \approx I$ which is termed as an attractor. The transformation, T , then forms the necessary data to code the image \hat{I} . The transformation is typically formed as a "collage" of smaller transformations, for different nonoverlapping regions of the image, termed as range blocks. Each range block is defined to be some transformation of a domain block taken from the same image. The transformation usually consists of a spatial scale (usually fixed at 2 : 1), rotation, reflection, spatial offset, contrast scaling and luminance offset. Thus the data to be coded consists of a range block partitioning and a set of transform parameters for each range block.

Although contrast scaling and luminance offset parameters for any candidate domain block can be found using a least squares approach, a fractal image coder still needs to search for the best domain block for each range block. This is typically performed using an exhaustive search method over all candidate domain blocks up to a given distance, making the task of encoding images computationally expensive. In order to reduce this complexity, Redmill and Bull have applied GA's to search for the best match domain block [28]. In initial experiments, a fixed range block partitioning into 8×8 pixel blocks was used, and the search space was limited to offsets only (no rotations or reflections). The absence of rotations and reflections was found to give better results for exhaustive search and allows the use of a multi-step algorithm similar to that used in motion estimation. The GA was implemented in a similar way to that described for motion estimation. The search space consists of two-dimensional integer vectors up to a pre-specified limit. Mutation was implemented as Gaussian-distributed mutation vectors which were added to the candidate displacement vector. The population was seeded with a sampled grid to provide a good coverage of the search space with a small population, and a smaller population size (about 80% of the initial population size) was used for subsequent generations. This allowed the GA to better compromise between the initial random search and subsequent exploitation. Good results were found after only a few (less than 10) generations.

Fig. 10 shows a comparison between the exhaustive search, multistep search, and GA approaches. Each evaluation consists of calculating the optimum contrast scaling and luminance offset parameters and the resulting MAD between the range block and transformed domain block. The results show that, despite the small population and number of generations, the GA exhibits reliable convergence and can outperform exhaustive search and heuristic methods. Again it is anticipated that GA's can be used with more complicated transformation models, for which simple heuristic methods don't exist. However, as before, the cost of transmission of the extra parameters must be taken into account.

VII. VECTOR QUANTIZATION

Vector quantization (VQ) [1], [29] maps a Euclidean space \mathbb{R}^l on to a finite subset V of \mathbb{R}^l , using a codebook optimized to give the minimum error between the original and reconstructed images. The encoder will select the nearest codebook entry to the current image block and transmit the address to the decoder, where it is used to address an identical codebook. VQ

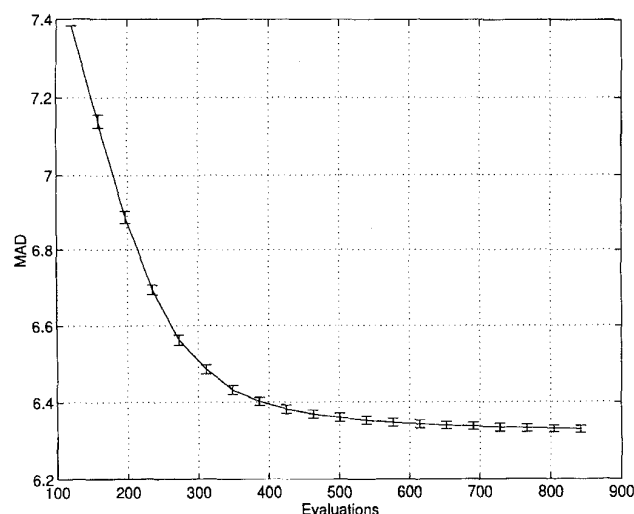


Fig. 10. Convergence of GA for fractal search.

has been widely investigated as a tool for image compression, both in isolation and in combination with other methods such as the DCT and subband coding. The most popular approaches to VQ codebook design are based on derivatives of the Linde, Buzo, and Gray (LBG) algorithm [29]. Although little work has been reported in this field, GA's clearly have potential in two distinct areas: codebook design and codebook search. Jiang and Butler [30] have recently presented some preliminary results in this area using GA's to design codebooks for 4×4 image blocks using a neural network-based VQ strategy. Results for a range of standard images are given, with comparisons between the GA approach and a single alternative competitive learning algorithm. Improvements in PSNR of up to 15% are claimed in this case. This work is clearly at an early stage but represents an application for GA's worthy of further investigation.

VIII. FEATURE- AND OBJECT-BASED IMAGE AND VIDEO CODING

In the context of low bit-rate coding schemes, there is a movement toward model-based approaches. These aim to describe an image in terms of the objects it contains rather than the values of regularly sampled picture elements. Approaches range from the use of wireframe models, to scene analysis methods based on segmentation in terms of intensity, texture, or motion characteristics. These methods require that the representative features can be accurately found in the image.

The problem of feature detection within images is essentially a computationally intensive search and classification problem, for which GA's have been found to be effective and are widely used. They have been successfully used for segmentation [31]–[34], edge detection [35], and feature detection [36], [37]. Toet *et al.* [38], for example, consider the use of GA's for contour matching between a reference image and a distorted or noisy version of the same. Scale and orientation independent results are presented for aerial reconnaissance images of military vehicles. Other applications include medical image segmentation and object recognition [39].

IX. CONCLUSIONS AND FUTURE DIRECTIONS

This paper has identified the potential and practical application of GA's in the design and realization of image and video coding systems and related applications. The results presented show that GA's are capable of robust performance, even with problems spaces which are discontinuous or deceptive. They have been shown to be capable of producing superior performance to conventional methods, often with similar or reduced computational effort.

Several areas exist for future study. For example, the combination of a GA with more localized optimization methods such as hill climbing or low-temperature simulated annealing has been shown, in other application areas, to speed up convergence and to yield improved final results. A more recent innovation, worthy of further study, has been the adoption of genetic programming (GP) techniques, to evolve both the structure and parameters for a signal processing function [40].

Nonlinear filtering is also an emerging area with significant promise in signal and image processing applications. However, the design of algorithms based on nonlinear operations (e.g., morphological filters) can be computationally demanding or even intractable. Recent work by Harvey and Marshall [41] has demonstrated the potential of GA's in this area.

Finally, we have seen that for many of the problems considered in this paper, that multiple objective criteria exist, which often conflict and rarely produce a unique solution. In most GA work multiple objectives are combined (usually linearly) to form a single fitness function. There is however an emerging trend toward truly multiobjective optimization using approaches based on Pareto-like ranking methods. Recent work on multiple criterion optimization using GA's [42] shows that GA's hold the promise for the automated design of systems which are globally optimum, not just in terms of algorithmic performance but also in terms of technological constraints and architectural efficiency.

REFERENCES

- [1] R. J. Clarke, *Digital Compression of Still Images and Video*. New York: Academic, 1995.
- [2] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation, Compression and Standards*. New York: Plenum, 1995.
- [3] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press, 1975.
- [4] D. Beasley, D. R. Bull, and R. R. Martin, "Genetic algorithms, pt. 1: An introduction," *Univ. Comput.*, vol. 15, no. 3, pp. 58–69, 1993.
- [5] D. Beasley, D. R. Bull, and R. R. Martin, "Genetic algorithms, pt. 2: Advanced techniques," *Univ. Comput.*, vol. 15, no. 4, pp. 170–181, 1993.
- [6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [7] D. R. Bull and D. H. Horrocks, "Primitive operator digital filters," *Proc. Inst. Elect. Eng.*, pt. g, vol. 138, no. 3, pp. 401–412, 1991.
- [8] Y. C. Lim and B. Liu, "Design of cascade form FIR filters with discrete valued coefficients," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1735–1739, 1988.
- [9] L. Banzato, "A design method for 2-D multiplierless FIR digital filters for video applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 273–284, 1992.
- [10] P. Siohan and A. Benlismane, "Finite precision design of optimal linear phase 2-D FIR digital filters," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 11–21, Jan. 1989.
- [11] J. D. Schaffer and L. J. Eshelman, "Designing multiplierless digital filters using genetic algorithms," in *Proc. 5th Int. Conf. Genetic Algorithms*, Urbana Champaign, IL, June 1993, pp. 439–444.
- [12] E. C. Ifeachor and S. Harris, "A new approach to frequency sampling digital filter design using genetic algorithms," in *Proc. IEE Workshop on Natural Algorithms in Signal Processing*, Essex, U.K., 1993, pp. 5/1–5/8.

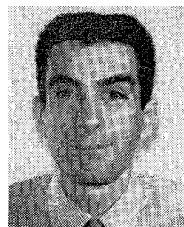
- [13] G. Wade, A. Roberts, and G. Williams, "Multiplierless FIR filter design using a genetic algorithm," in *Proc. Inst. Elect. Eng. (Vis. Image Signal Process)*, vol. 141, no. 3, June 1994, pp. 175–180.
- [14] J. G. McWhirter, "Algorithmic engineering in adaptive signal processing," in *Proc. Inst. Elect. Eng.*, pt. f, vol. 139, no. 3, June 1992, pp. 226–232.
- [15] D. R. Bull, G. Wacey, J. J. Stone, and J. Soloff, "A compound primitive operator approach to the realization of video sub-band filter banks," in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing*, 1993, pp. 405–408.
- [16] D. R. Bull and A. Aladjidi, "The optimization of multiplier-free directed graphs: An approach using genetic algorithms," in *Proc. 1994 IEEE Int. Symp. on Circuits and Systems*, London, 1994, pp. 456–459.
- [17] S. Sriranganathan, D. R. Bull, and D. W. Redmill, "Design of 2-D multiplierless FIR filters using genetic algorithms," in *Inst. Elect. Eng. Int. Conf. Genetic Algorithms in Engineering Systems*, Sheffield, U.K., 1995, pp. 282–286.
- [18] S. Sriranganathan, D. R. Bull, and D. W. Redmill, "Optimization of multiplierless 2-D digital filters," in *VCIP-96*, to be published.
- [19] D. B. Harris and R. M. Merserau, "A comparison of algorithms for minimax design of two-dimensional linear phase FIR filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 492–500, Dec. 1977.
- [20] H. Gharavi, "Subband coding algorithms for video applications: Video-phone to HDTV conferencing," *IEEE Trans. Veh. Technol.*, vol. 1, pp. 174–183, 1991.
- [21] J. M. Fitzpatrick, J. J. Grefenstette, and D. Van Gucht, "Image registration by genetic search," in *IEEE Southeast Conf.*, 1994, pp. 460–464.
- [22] J. J. Grefenstette and J. M. Fitzpatrick, "Generic search with approximate function evaluations," in *Proc. Int. Conf. on Genetic Algorithms and Their Applications*, 1985, pp. 112–120.
- [23] ———, "Genetic algorithms in noisy environments," *Machine Learning*, vol. 3, pp. 101–120, 1988.
- [24] J. J. Jacq and C. Roux, "Registration of 3-D images by genetic optimization," *Pattern Recognition Lett.*, vol. 16, pp. 823–841, 1995.
- [25] P. W. M. Tsang, "A genetic algorithm for affine invariant object shape recognition," in *Inst. Elect. Eng. Int. Conf. Genetic Algorithms in Engineering Systems*, Sheffield, U.K., 1995, pp. 293–298.
- [26] J. F. Vega-Riveros and K. Jabbar, "Review of motion analysis techniques," in *Proc. Inst. Elect. Eng.*, vol. 136, 1989, pp. 397–404.
- [27] A. E. Jaquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Processing*, pp. 8–30, 1992.
- [28] D. W. Redmill, D. R. Bull, and R. R. Martin, "Genetic algorithms for fast fractal search in fractal image coding," *SPIE Conf. Visual Communications and Image Processing*, Orlando, FL, 1996, to be published.
- [29] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantiser design," *IEEE Trans. Communicat.*, vol. 28, pp. 84–95, 1980.
- [30] J. Jiang and D. Butler, "A genetic algorithm design for vector quantization," in *Inst. Elect. Eng. Int. Conf. Genetic Algorithms in Engineering Systems*, Sheffield, U.K., 1995, pp. 331–336.
- [31] B. Bhanu, S. Lee, and S. Das, "Adaptive image segmentation using genetic algorithms and hybrid search methods," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, pp. 1268–1291, 1995.
- [32] S. Cagnoni, A. B. Dobreniecki, J. C. Yanch, and R. Poli, "Interactive segmentation of multi-dimensional medical data with contour based application of genetic algorithms," in *IEEE Int. Conf. Image Processing*, Austin, TX, vol. III, 1994, pp. 498–502.
- [33] P. Andrey and P. Tarroux, "Unsupervised image segmentation using a distributed genetic algorithm," *Pattern Recognition*, vol. 27, pp. 659–673, 1994.
- [34] X. Tu, T. D. Bui, and A. Kryzak, "Robust estimation for range segmentation and reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 530–538, 1994.
- [35] S. M. Bhandarkar, Y. Zhang, and W. D. Potter, "An edge detection technique using genetic algorithm-based optimization," *Pattern Recognition*, vol. 27, pp. 1180–1194.
- [36] M. S. White, E. E. Davies, and S. J. Flockton, "The application of genetic algorithms to feature recognition and object counting," in *Inst. Elect. Eng. Colloquium on Genetic Algorithms in Image Processing and Vision*, London, U.K., Oct. 1994.
- [37] J. E. Smith, T. C. Fogarty, and I. R. Johnson, "Genetic selection of features for clustering and classification," in *Inst. Elect. Eng. Colloquium on Genetic Algorithms in Image Processing and Vision*, London, U.K., Oct. 1994.
- [38] A. Toet, "Genetic contour matching," in *Proc. Inst. Elect. Eng. Workshop on Natural Algorithms in Signal Processing*, Essex, U.K., 1993, pp. 9/1–9/8.
- [39] K. Delibasis and P. E. Undrill, "Genetic algorithms and deformable geometric models for anatomical object recognition," *Inst. Elect. Eng. Colloq. Genetic Algorithms in Image Processing and Vision*, London, U.K., pp. 8/1–8/7, Oct. 1994.
- [40] K. C. Sharman, A. Esparcia Alcazar, and Y. Li, "Evolving signal processing algorithms by genetic programming," *Inst. Elect. Eng. Int. Conf. Genetic Algorithms in Engineering Systems*, Sheffield, U.K., pp. 473–480, 1995.
- [41] N. R. Harvey and S. Marshall, "Using genetic algorithms in the design of morphological filters," *Inst. Elect. Eng. Colloq. Genetic Algorithms in Image Processing and Vision*, London, U.K., pp. 6/1–6/5, Oct. 1994.
- [42] C. M. Fonseca and P. J. Fleming, "Multiobjective genetic algorithms made easy: Selection sharing and mating restriction," in *Proc. Inst. Elect. Eng. Int. Conf. Genetic Algorithms in Engineering Systems*, Sheffield, U.K., 1995, pp. 45–52.



David R. Bull (S'83–M'83) received the B.Sc. degree in 1980, the M.Sc. degree in 1983, and the Ph.D. degree in reduced complexity digital signal processing from the University of Wales in 1998.

He is currently a Reader in Digital Signal Processing in the Department of Electrical and Electronic Engineering at the University of Bristol where he leads the Image Communications Group. Prior to this appointment he has held posts as a lecturer at The University of Wales, Cardiff, and as an Electronic Systems engineer for Rolls-Royce. His research interests include theoretical, architectural, and application aspects of digital signal processing, in particular image and video coding techniques for wireless communications. In the context of this work he has been investigating the optimization of signal processing algorithms and architectures using evolutionary techniques. He holds several research grants and has acted as a consultant to numerous companies in these areas. He has published more than 80 papers in learned journals and conferences.

Dr. Bull was a recent recipient of the Institution of Electrical Engineers Ambrose Fleming Premium Award for his work on primitive operator digital filters.



David W. Redmill received the B.A. (hon.) degree in electrical and informational sciences in 1991, and the Ph.D. degree in 1995, both from the Department of Engineering at the University of Cambridge, U.K.

Since then he has been a Research Assistant with the Image Communications Group in the Center for Communications Research at University of Bristol. His research interests cover the areas of image and video coding with particular emphasis on algorithm and architecture optimization using GA's, error resilient image coding, and novel filter-bank methods for image compression.